# A Power Management Method to Improve Energy Budget Utilization

Sho Ishii
Graduate School of Information Sciences, Tohoku University
Sendai, Miyagi, Japan

Keichi Takahashi
Cyberscience Center, Tohoku University
Sendai, Miyagi, Japan

Yoichi Shimomura
Cyberscience Center, Tohoku University
Sendai, Miyagi, Japan

Hiroyuki Takizawa
Cyberscience Center, Tohoku University
Sendai, Miyagi, Japan

## 1 INTRODUCTION

The power consumption of High Performance Computing (HPC) systems has been inevitably increasing with their computing power. Due to the fact that IT infrastructure and power demand are becoming economically unaffordable, it is anticipated that future HPC systems will have a set energy budget [2]. In light of the above situation, this study focuses on *energy budget* as an indicator in power management.

Previous work [1] identified that improving the energy budget utilization can increase the system throughput. Given this benefit, we aim to improve the utilization of energy budget and increase the system throughput in this study.

## 2 PROPOSAL

We propose a power management method to improve energy budget utilization. The power cap of each job is adjusted based on the surplus of energy budget generated by low-load jobs. The surplus energy is later consumed by high-load jobs to increase the energy budget utilization. The proposed method keeps track of the *surplus energy*, *i.e,* the cumulative difference between the energy allocated to jobs and the energy actually consumed by the jobs. The surplus energy is defined by the following equation:

$$S_j = PC_{\mathrm{default}} T_j + S_{j-1} - E_j, \tag{1}$$

where $S_j$ is the surplus energy on a node at the end of job $j$. Note that we assume that only one job is executed on a node at a time. $PC_{\mathrm{default}}$ is the amount of power that could be constantly consumed if the power cap was fixed, *i.e.*, the energy budget for a given period of time divided by the duration. $T_j$ and $E_j$ are the total runtime and energy consumption of job $j$, respectively. Since the power consumption of a job never exceeds its power cap, $PC_{\mathrm{default}} T_j + S_{j-1}$ is the maximum energy that job $j$ can consume.

Using the surplus energy available at that time, the following equation calculates the power cap enforced to a job:

$$PC_{j+1} = \min\left(PC_{\mathrm{default}} + \frac{S_j}{\hat{T}_{j+1}}, PC_{\mathrm{max}}\right), \tag{2}$$

where $\hat{T}_{j+1}$ is the predicted runtime of job $j+1$. In practice, the job runtime requested by the user could be used instead, even though it is inaccurate in most cases. $PC_{\mathrm{max}}$ is the maximum power cap supported by the target system.

Since an HPC system consists of many nodes, we compare our method when sharing and not sharing surplus energy among nodes. When the surplus energy is not shared among nodes, it is calculated on a per-node basis. On the other hand, when the surplus energy is shared among nodes, it is calculated for each node, and then the average surplus energy over all nodes is redistributed to each node.

## 3 EVALUATION

To evaluate the effectiveness of the proposed method, we compare the proposed method with a baseline using a fixed power cap with our own simulator. The simulated cluster consists of 200 compute nodes.

**Table 1: Results with the same number of compute-intensive and memory-intensive jobs.**

| Method | Total energy consumption | Energy budget utilization | Makespan |
|---|---:|---:|---:|
| Proposed (w/o sharing) | 1.079 GJ | 93.57% | $9.36 \times 10^4$s |
| Proposed (w/ sharing) | 1.116 GJ | 98.62% | $8.70 \times 10^4$s |
| Baseline | 0.863 GJ | 68.11% | $9.75 \times 10^4$s |

Table 1 shows the simulation results. Compared to the baseline method where the power cap is fixed to $PC_{\mathrm{default}}$, the proposed method reduces the makespan by 9.0% and increases the energy budget utilization by 25.5% when surplus energy is not shared among nodes. When sharing surplus energy among nodes, the makespan is reduced by 10.7% and the energy budget utilization is increased by 30.5%. Accordingly, these results clearly show that the proposed method can properly adjust the power cap of each job so that the surplus energy generated by low-load jobs is used for accelerating high-load jobs, resulting in a higher energy budget utilization and a shorter runtime.

In the future, we will evaluate our method in an online scheduling scenario, where jobs are submitted according to Poisson arrival. We will also extend our energy management method to consider jobs waiting in the job queue.

## REFERENCES

[1] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Power Provisioning for a Warehouse-Sized Computer. In *Proceedings of the 34th Annual International Symposium on Computer Architecture.* 13–23.

[2] Srinivasan Ramesh, Swann Perarnau, Sridutt Bhalachandra, Allen D. Malony, and Pete Beckman. 2019. Understanding the Impact of Dynamic Power Capping on Application Progress. In *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS).* 793–804.